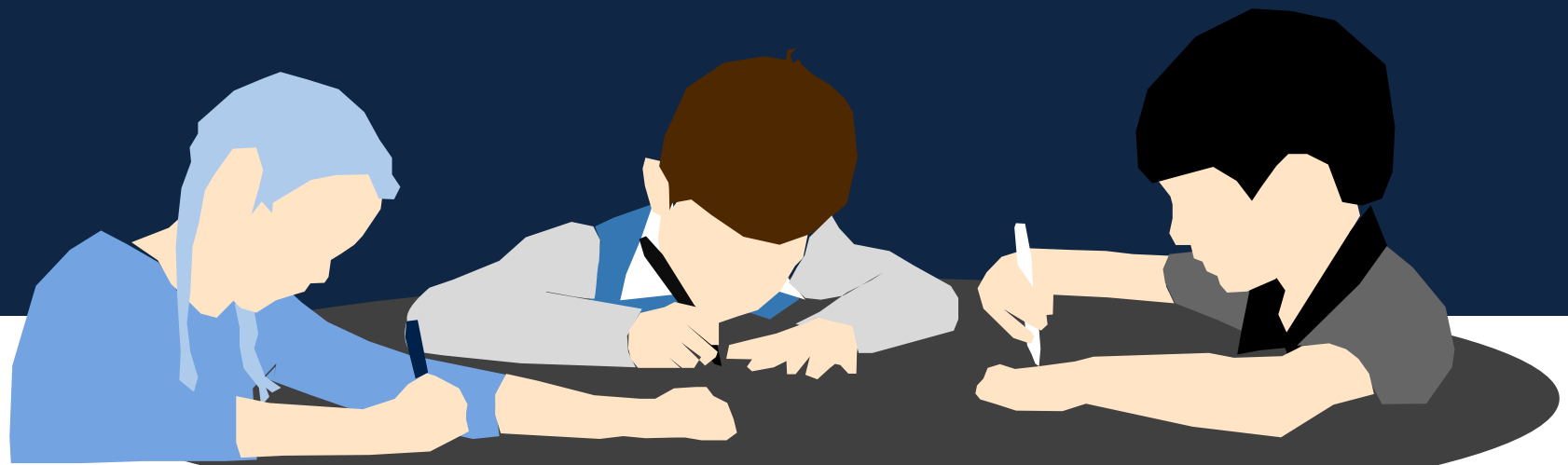
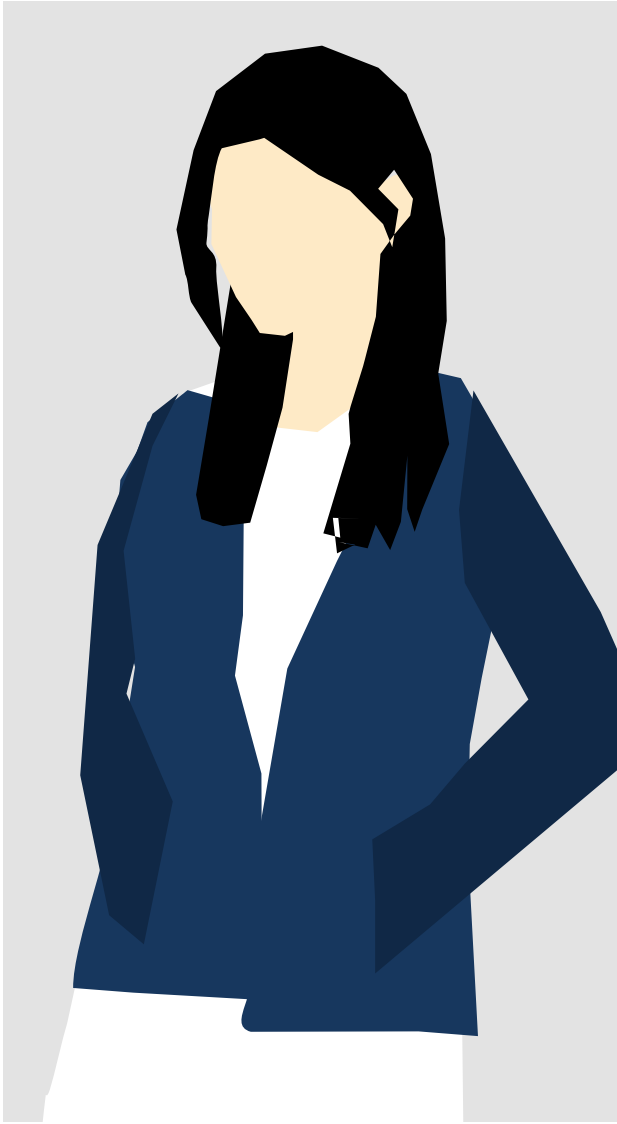
SWAN DELTA 2019

Anonymous peer feedback for problem-solving portfolios

Simon James

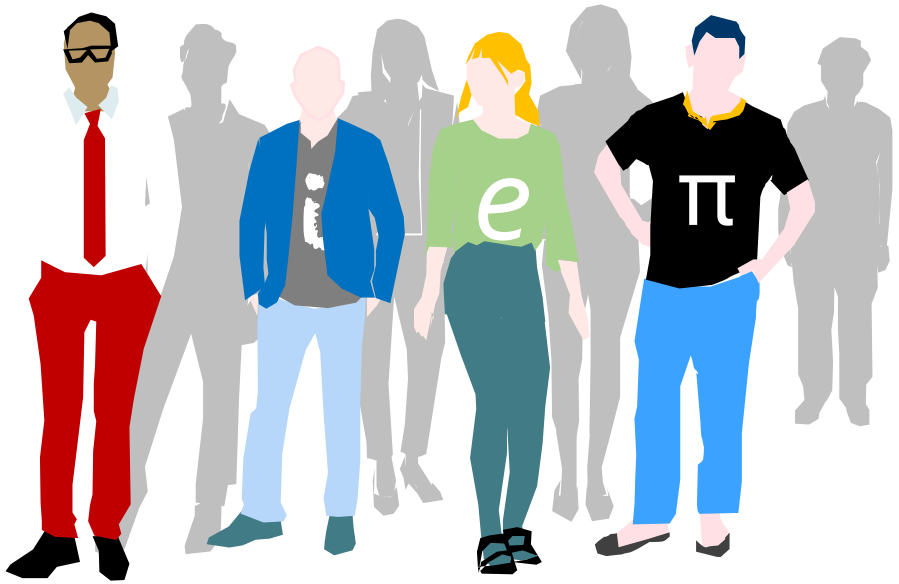
School of IT, Deakin University





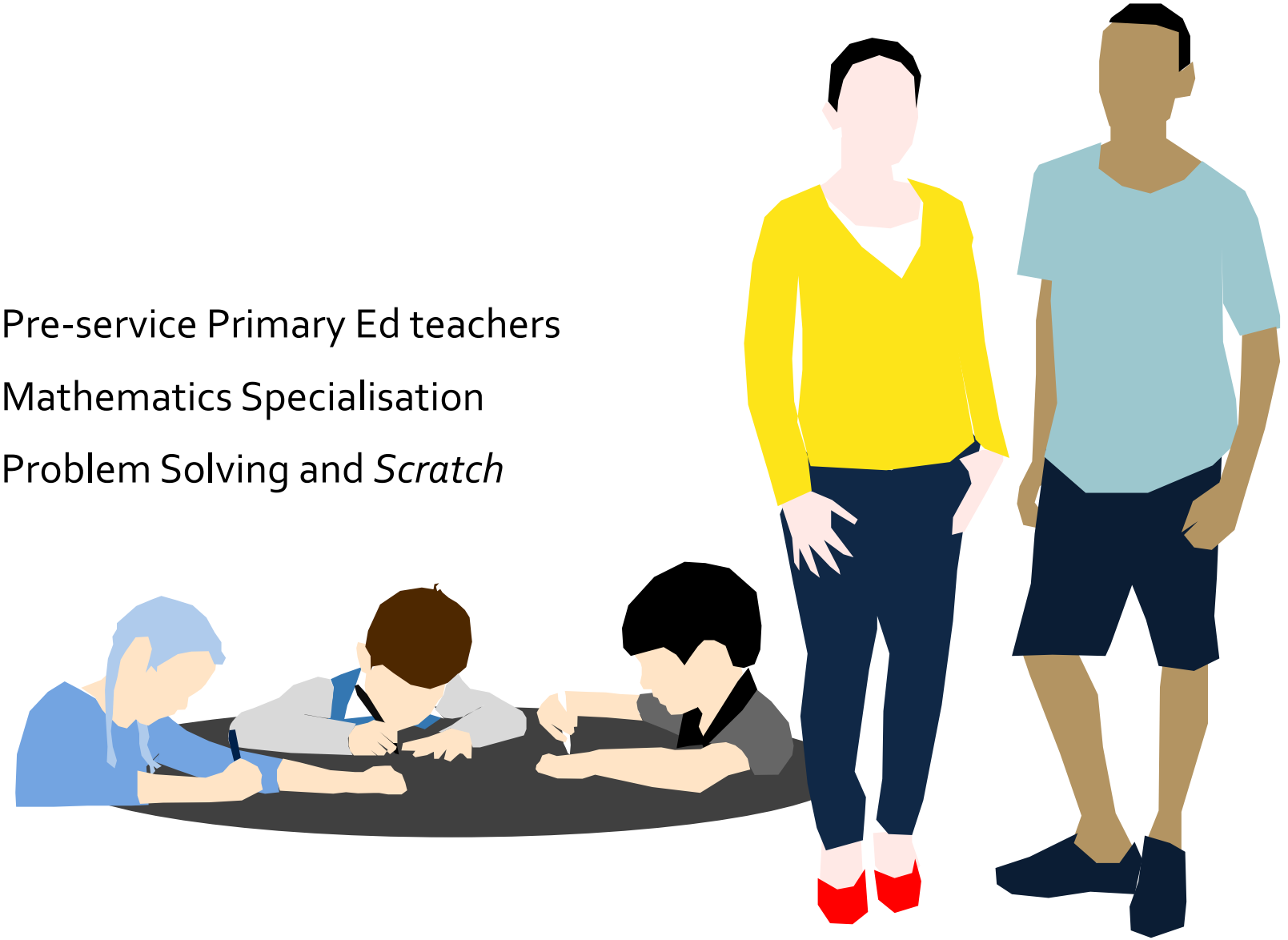
overview

- MOTIVATION
- PROCESS
- HOW IT WENT



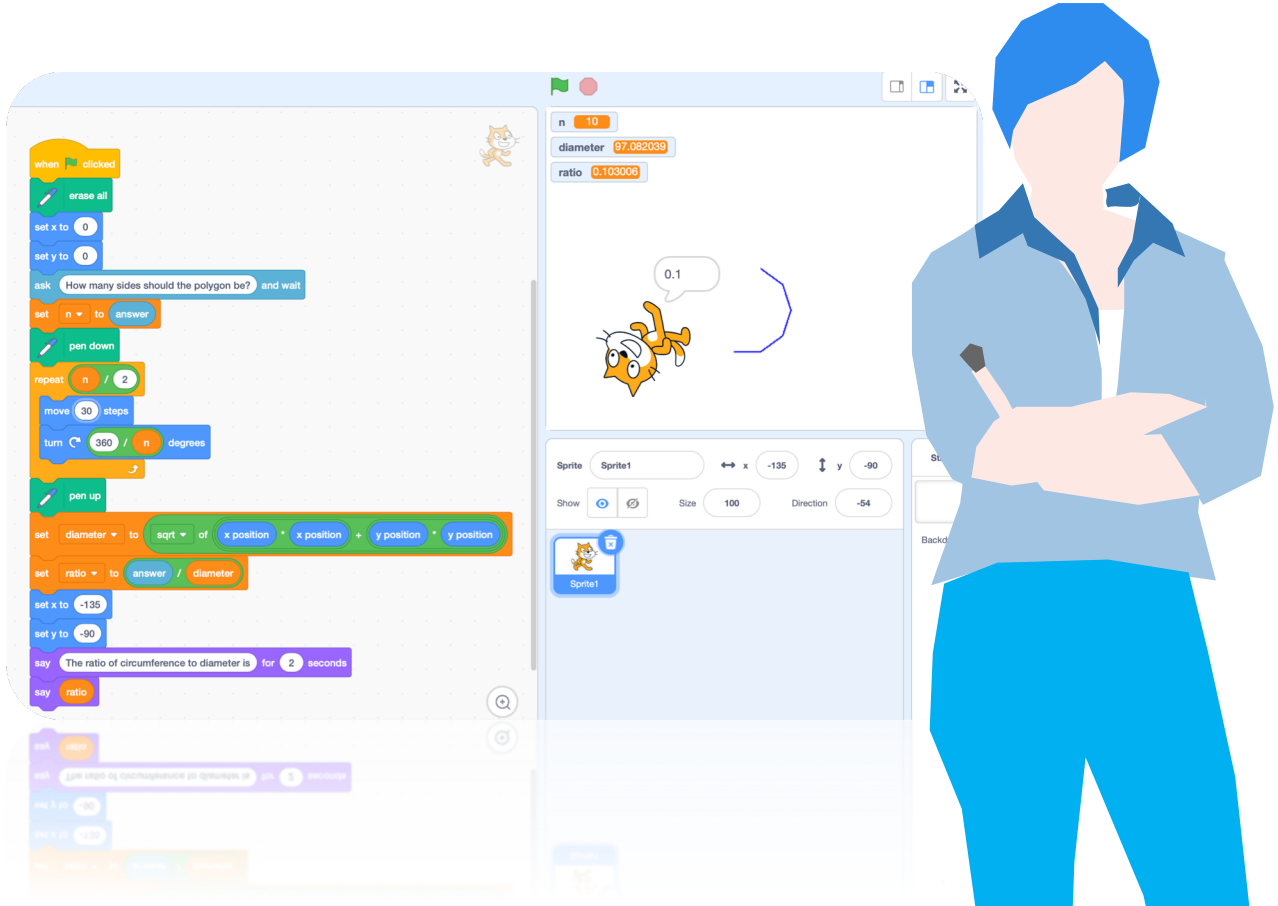
Context

- Pre-service Primary Ed teachers
- Mathematics Specialisation
- Problem Solving and *Scratch*



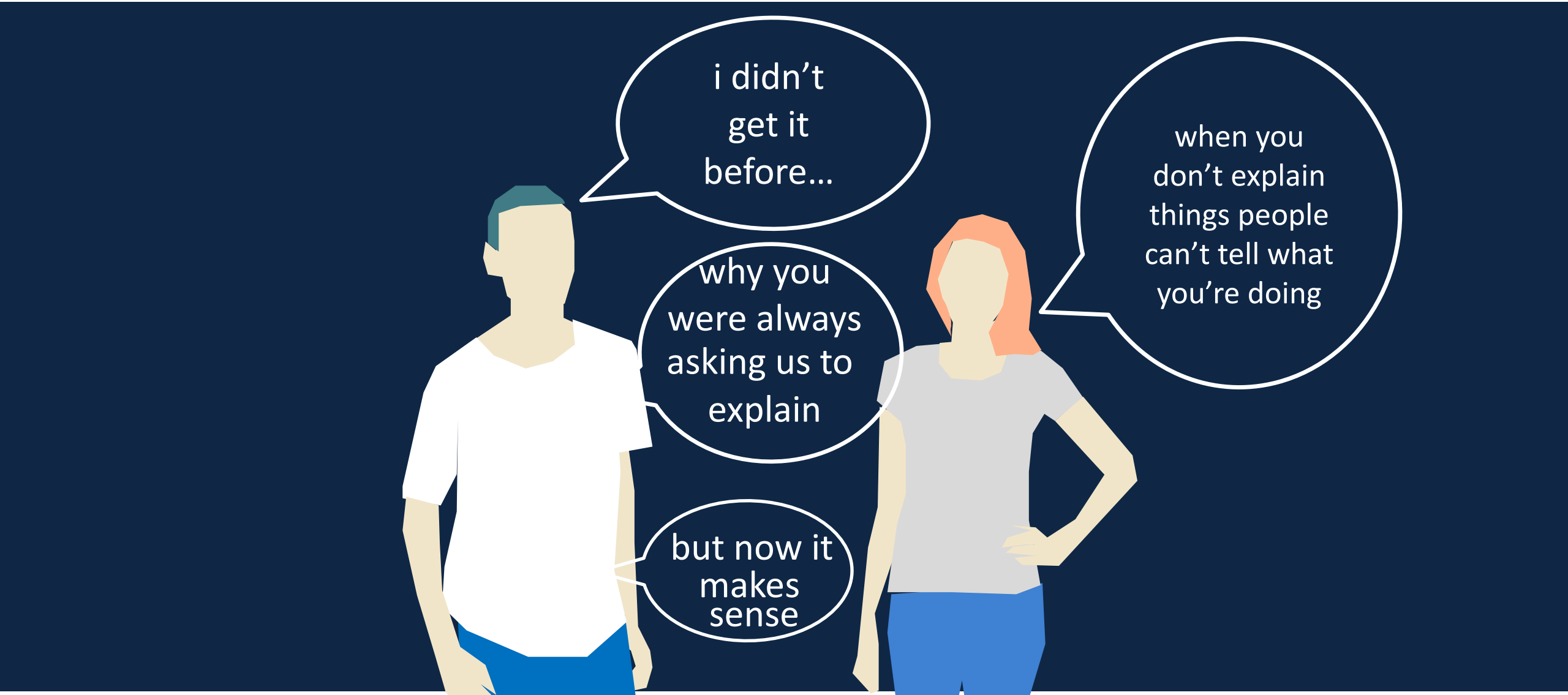
Why ?
(the first time)

- Assessing work can help consolidate understanding
- Academic time constraints
- Some assignments deserve to be shared



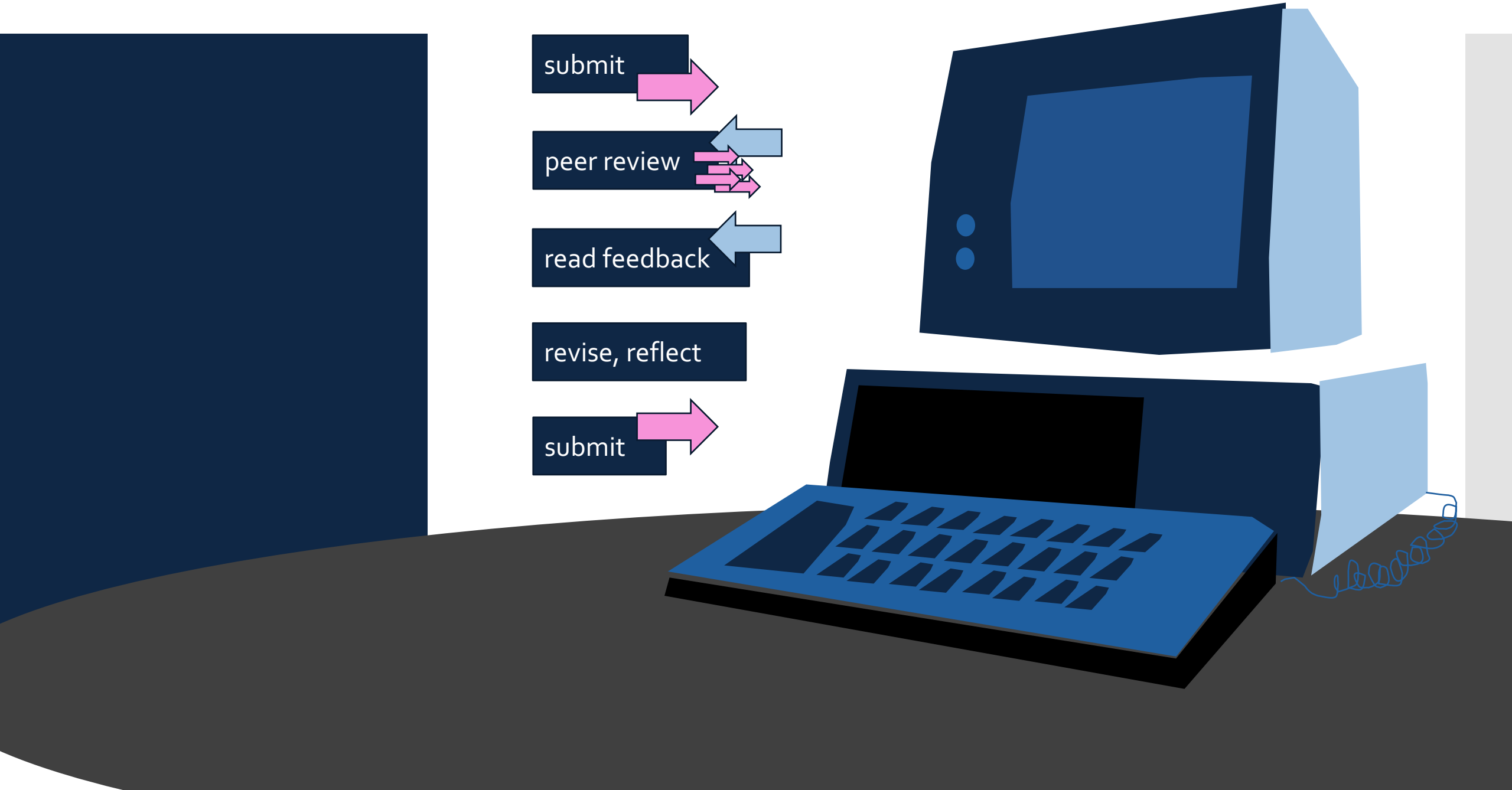
Why? (the second time)?

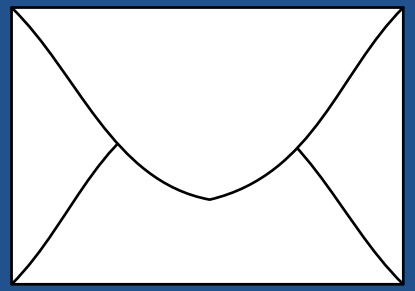
- Allow students to "do something" with what they learned in the first round of feedback
- Reinforce assignment aims and requirements
- De-emphasise the idea of a "single solution" in problem solving
- Encourage students to see communication as important



REINHOLZ, 2015


process
submit
peer review
read feedback
revise, reflect
submit


submit
peer review
read feedback
revise, reflect
submit



To mark	Link
200156	https://deakin365-my.sharepoint..
136459	https://deakin365-my.sharepoint..
205456	https://deakin365-my.sharepoint..
100166	https://deakin365-my.sharepoint..
108156	https://deakin365-my.sharepoint..

Section 1 of 2

Evaluations

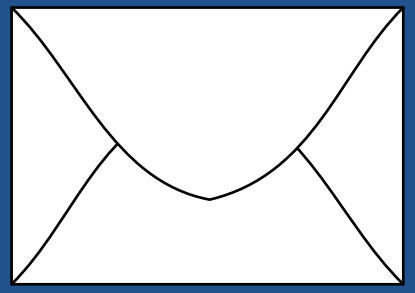
Description (optional)

Rubric and comments *

	N	N/P	P	P/C	C	C/D	D	D/HD	HD	HD+
Progra...	○	○	○	○	○	○	○	○	○	○
Incorpor...	○	○	○	○	○	○	○	○	○	○
Coding	○	○	○	○	○	○	○	○	○	○
Present...	○	○	○	○	○	○	○	○	○	○

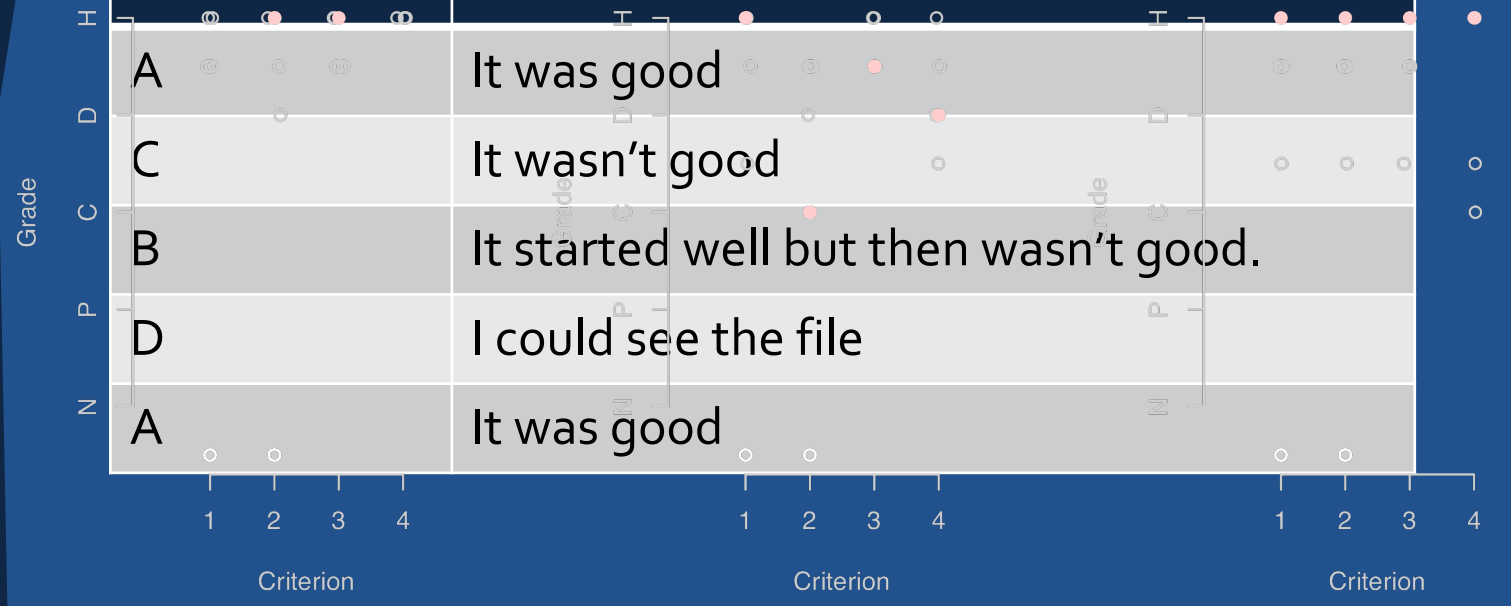
Please provide constructive comments that could help your peer improve their work, or be specific about aspects of the project they did well. *

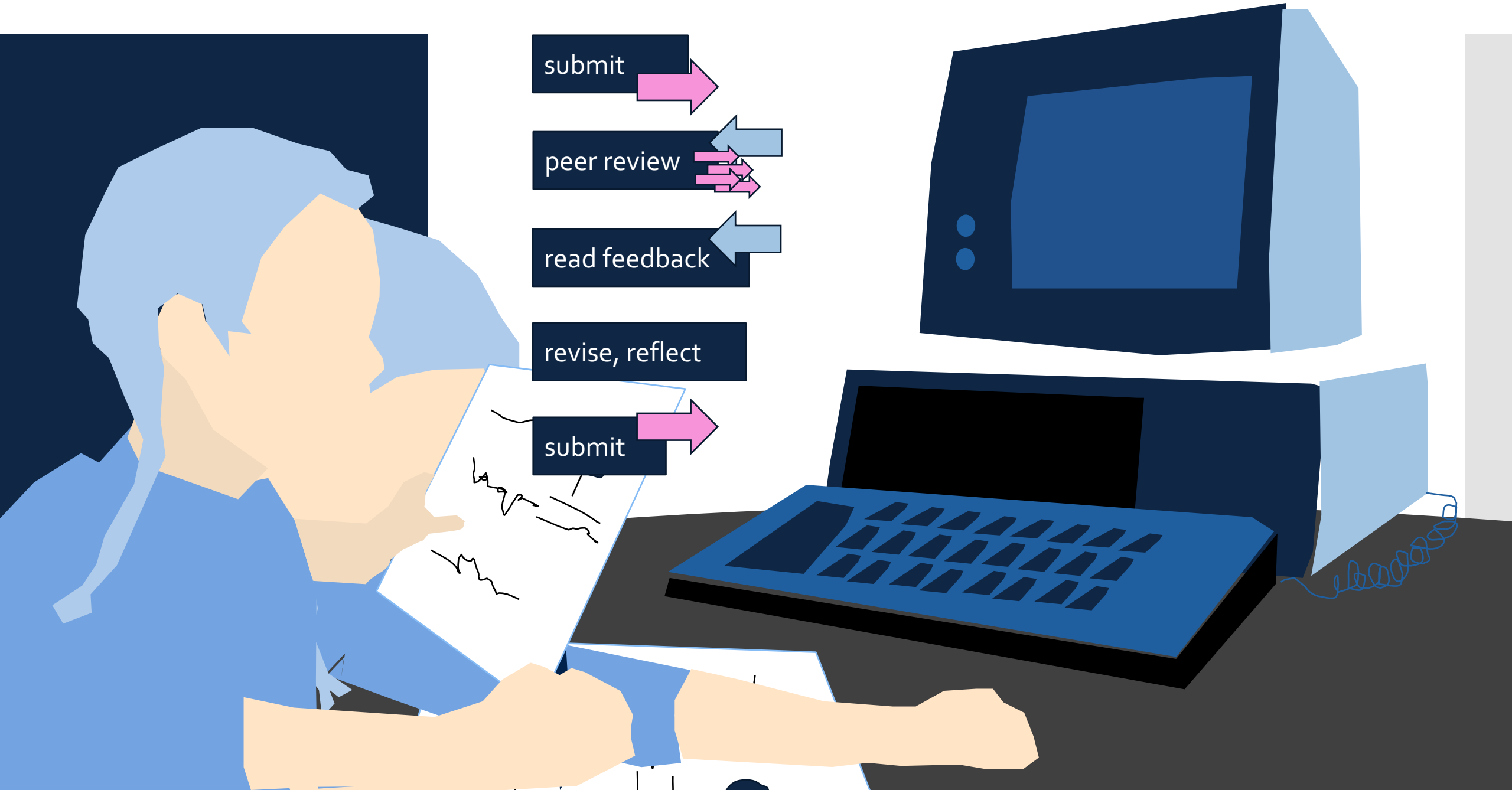
Long answer text



Student 220315106 Student 270300432 Student 260322522

Grades	Comment
A	It was good
C	It wasn't good
B	It started well but then wasn't good.
D	I could see the file
A	It was good



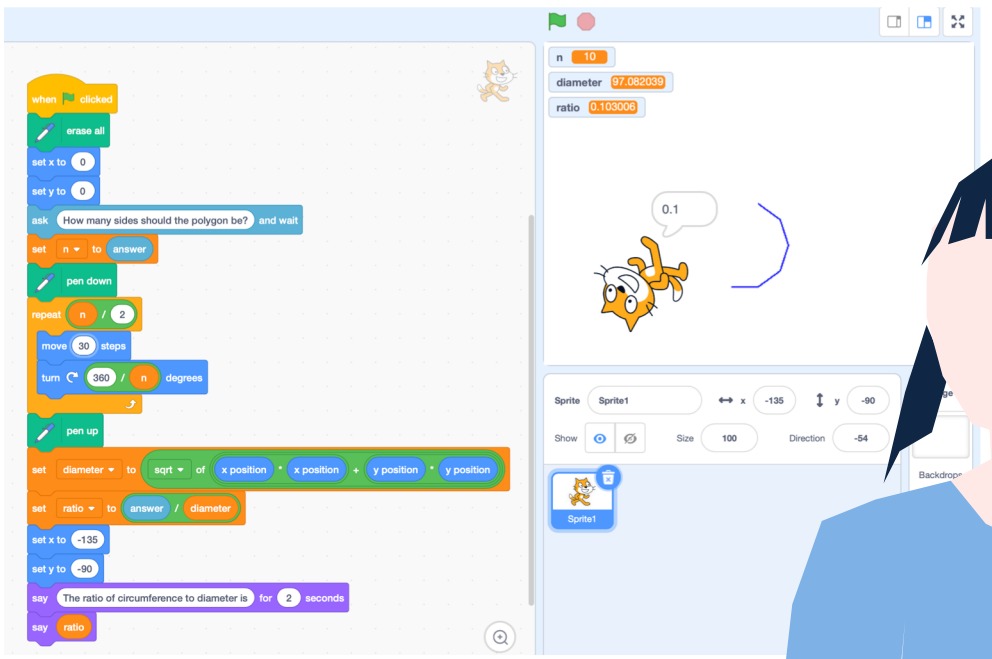

submit
peer review
read feedback
revise, reflect
submit

Coding Assignment

- **Task:** Design a program equivalent to about 20 lines that includes an if(…) command, a loop, at least one variable, and response to user input.

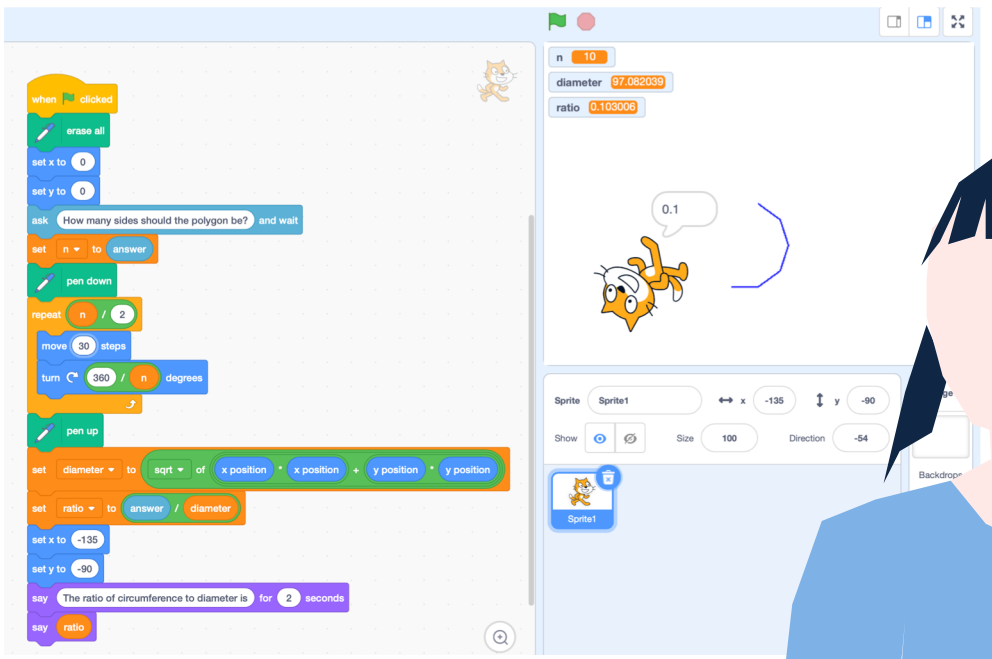
Criterion	Grade				
	N	P	C	D	HD
1. Program includes mandatory features	None of the required features included.	Program includes at least two of the required features and is not less than 15 lines.	Program incorporates some of the required features (if-statement, a loop, a variable, user input) but not all. Program is of an adequate length.	Program incorporates most of the required features (if-statement, loop, a variable, user input) but not all, or does not combine them to perform a meaningful task.	Program incorporates an if-statement, a loop, at least one variable, and responds to user input to achieve a meaningful task.
2. Incorporation of mathematics, originality, sophistication	Program doesn't run.	Program runs but only does something simple or similar to an example given during the course.	Program is interesting and relevant to a mathematical concept.	Program solves an interesting mathematical problem and is itself an interesting activity. Some degree of complexity involved in putting together functions we have studied in an original way.	Program solves an interesting mathematical problem or is itself an interesting activity, with some degree of originality and innovativeness in addressing the task
3. Coding	Code not provided.	Most coding features used correctly but overall the code does not quite achieve what it is intended for.	Program does not account for all possibilities of user input (or has no user input) and may include some mistakes.	Program accounts for all possibilities of user input but could easily be made more efficient or succinct (e.g. repeated actions rather than use of a loop)	Program accounts for all possibilities of user input and seems to be sensible and logical, with no obvious sections that could be improved with alternative syntax.
4. Presentation	Not provided, or not relevant to the assessment.	Adequate presentation although it is not overly clear what the code actually does.	Presentation demonstrates what the code does but without a lot of attention to engaging features.	Interesting presentation with clear description of what the code does.	Engaging presentation with a clear description of what the code does, conveying why it is interesting/useful and detailing any problems that had to be overcome or innovative things that were introduced.

Advice to students for providing feedback



- The marking rubric gives a general indication of the standard, your peer's work might differ in terms of how it meets any of the standards, so you need to use your evaluative judgement to estimate where it falls.

Advice to students for providing feedback

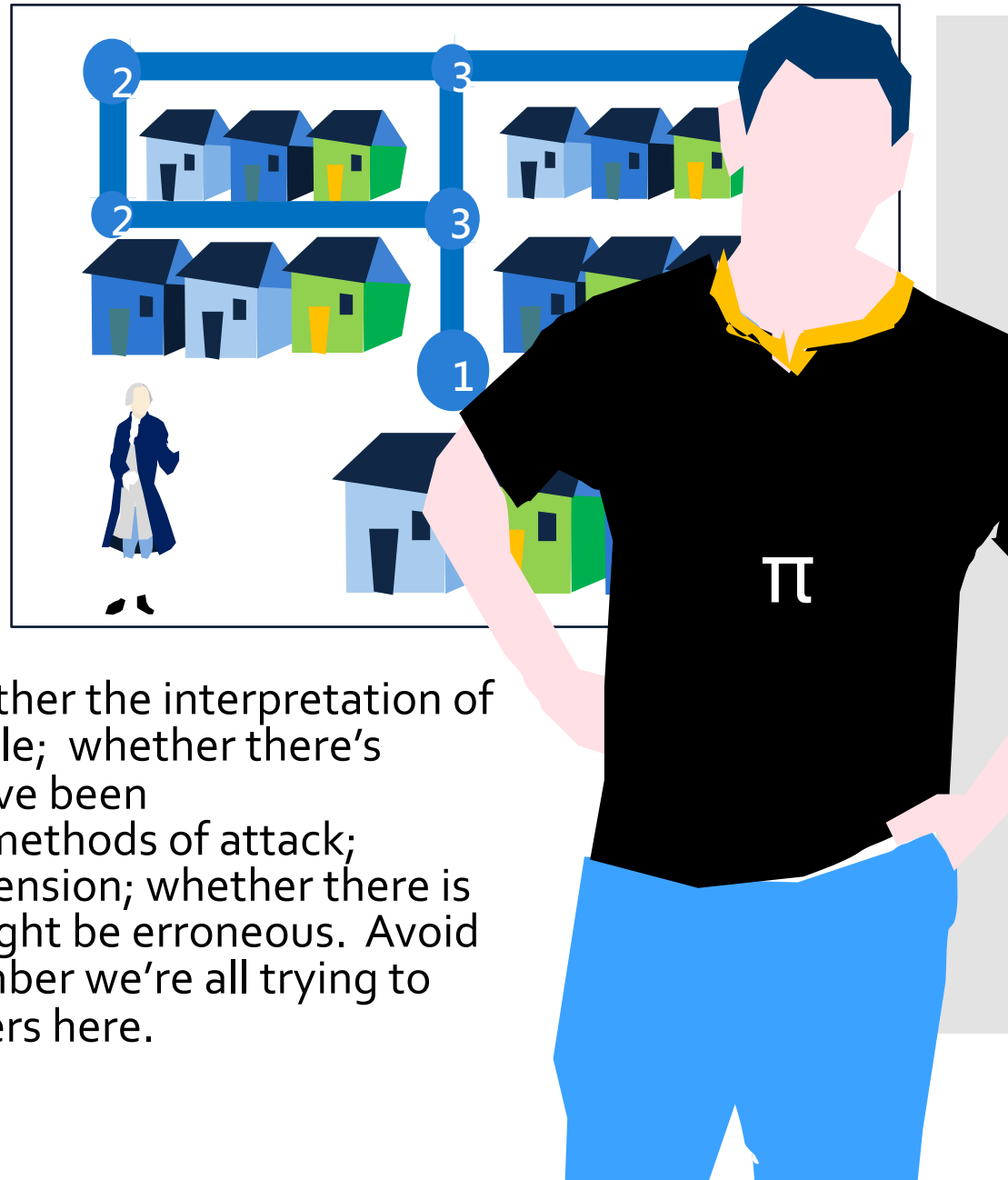


- For the comment, try to be specific and constructive. Usually students are interested to know why it has been marked lower or higher on any of the criteria, and what they could do differently. This will be especially helpful since you'll have an opportunity to update before final submission. It only needs to be 1-2 sentences.

Problem Solving Portfolios

- **Problem solving standard:** Satisfactory attempt and detail provided at entry, attack and review stages. Ambiguities and misunderstandings were clarified or addressed as part of engaging with the problem. Examples of specialisation/generalisation, conjecturing/justifying given. Opportunities for extension identified.

Advice to students for providing feedback

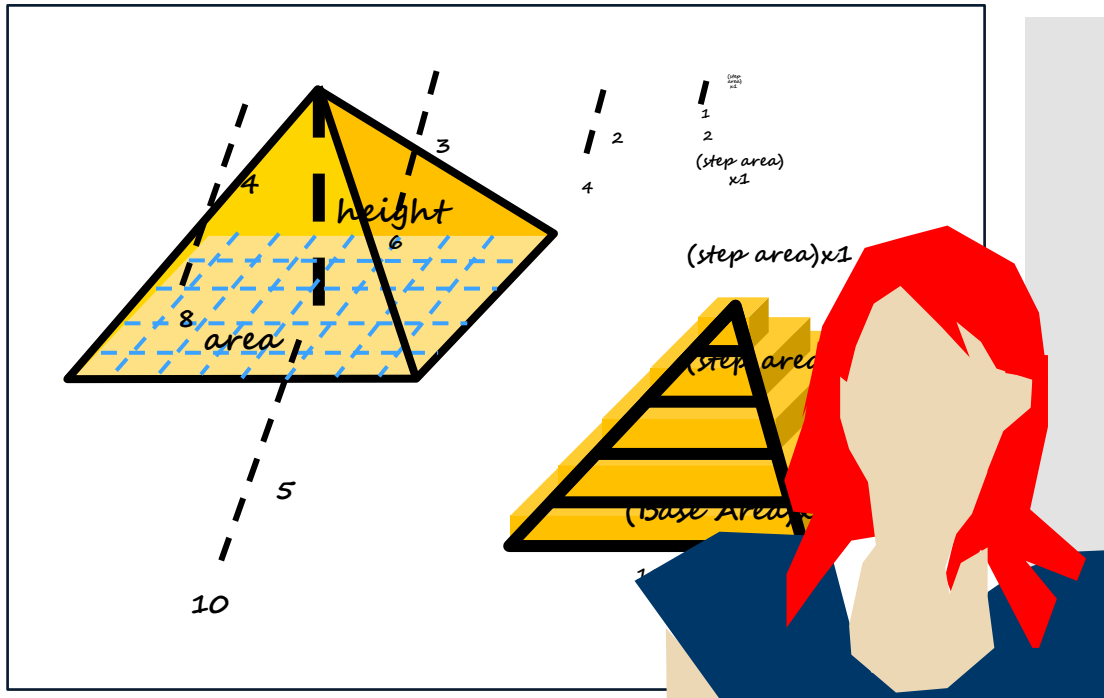


- So you can comment on whether the interpretation of the question seems reasonable; whether there's something that might not have been considered; other potential methods of attack; suggestions for review or extension; whether there is some reasoning used that might be erroneous. Avoid providing an answer – remember we're all trying to become better problem solvers here.

Problem Solving Portfolios

- **Instructional summary standard:** Clearly presented instructional summary, complete with simple examples and explanation that would make the concept understandable to someone unfamiliar with the topic.

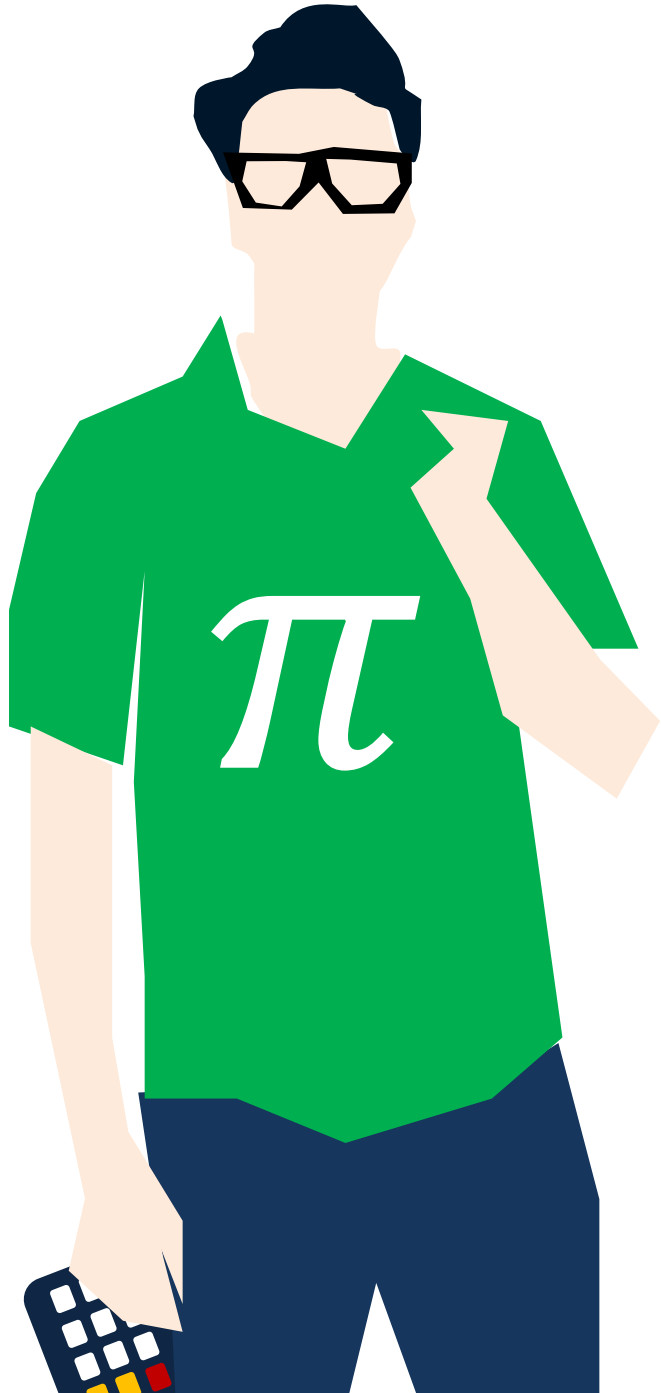
Advice to students for providing feedback



- So you can focus on the clarity of what is presented, whether an example or diagram might help, and whether there some to be any misunderstandings shown.

How did it go?

- Students tended to be good at pointing out requirements in terms of structuring problem solving solutions, making their explanations clear – less good at being able to get students "unstuck"
- Students gave a lot of positive encouragement
- There were only a few cases where the quality of feedback a student received overall was unhelpful



STUDENT FEEDBACK

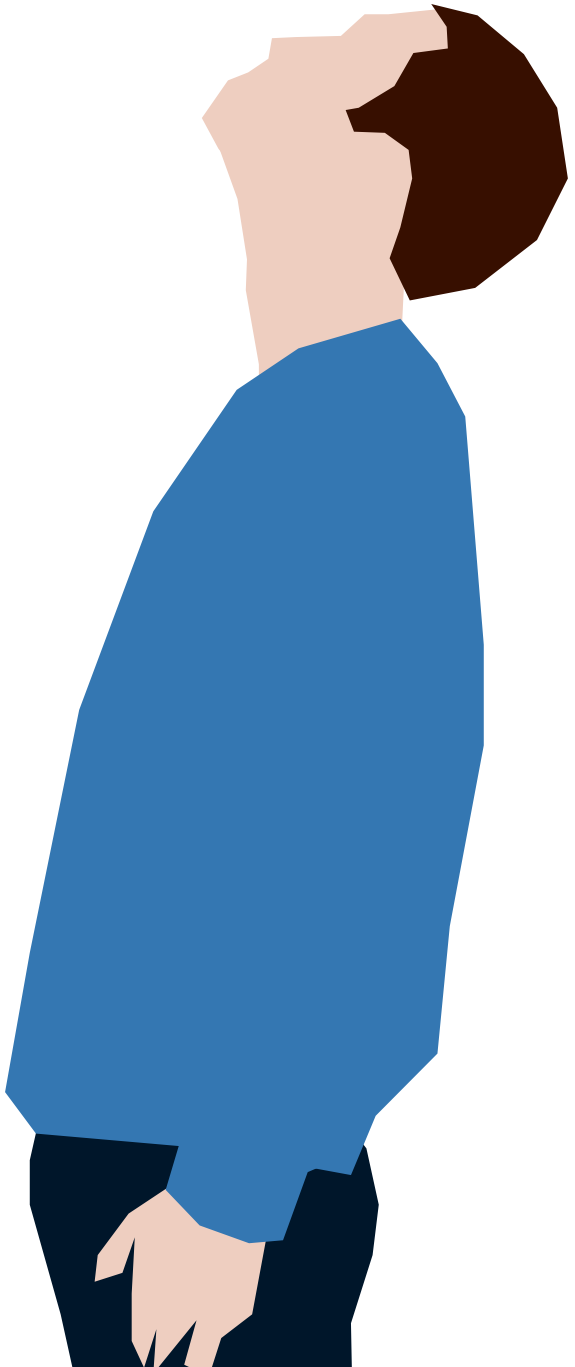
What did you like most about this unit?

- I really enjoyed evaluating other peoples work and seeing the different ways people think.
- Peer feedback was awesome
- I got a lot out of the peer reflection…
- Student feedback, it was great looking over other peoples work and seeing how they intemperate the problem and giving them advice to build upon their projects.
- The peer feedback helped me to improve my assignments. I learnt how to provide feedback which is an important aspect of teaching.
- Peer review aspect of assignments was great!

[Deakin ethics 2019-423]

future things…

- "Teach" feedback
  - Sample solutions (incorrect/correct)
  - Linking feedback to what was studied
  - Emotional responses to feedback
- Feedback Fruits



discussion?